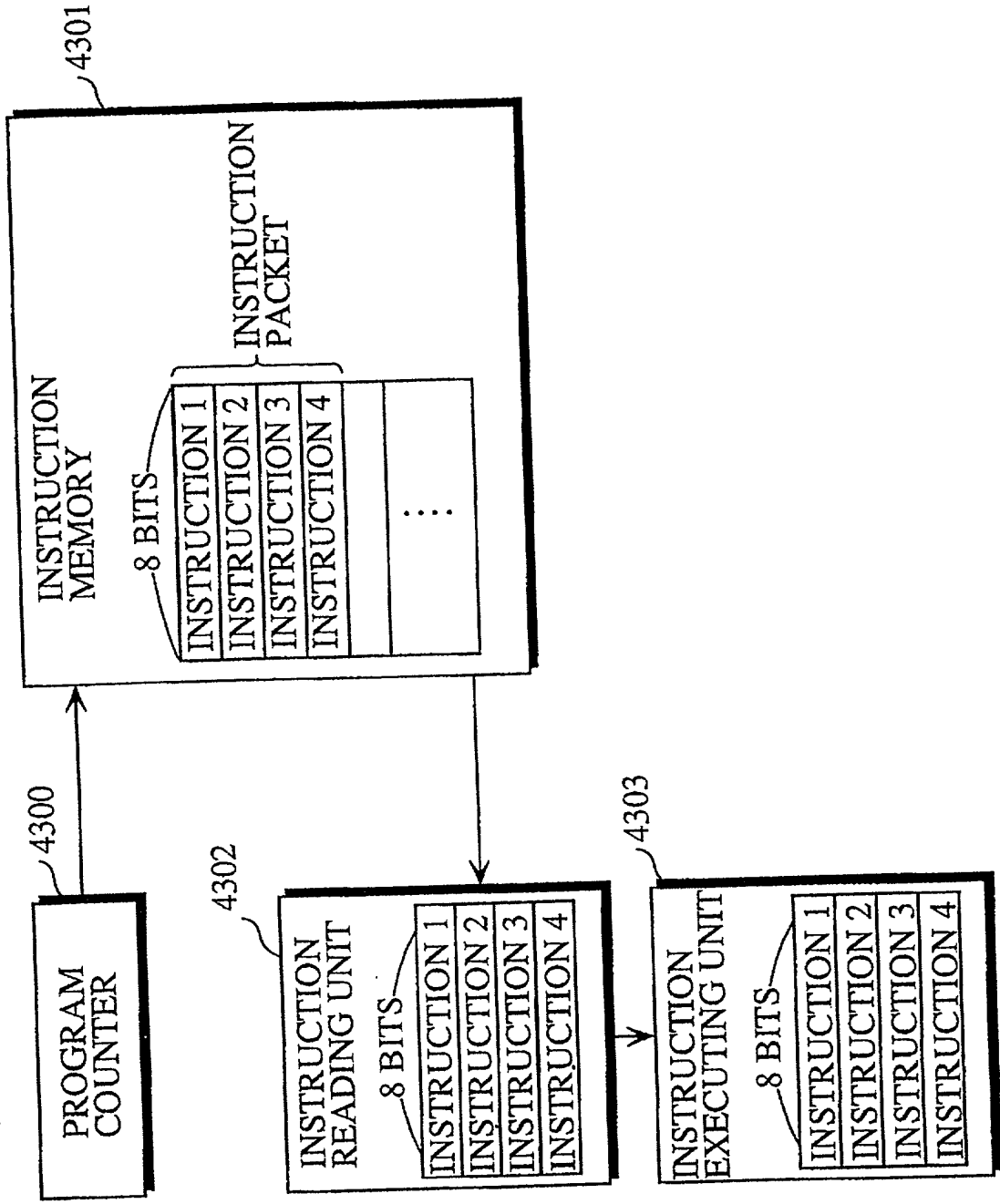


FIG. 1



PARALLEL EXECUTION BOUNDARY INFORMATION 100

FORMAT INFORMATION 101

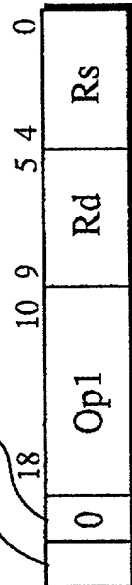


FIG. 2A

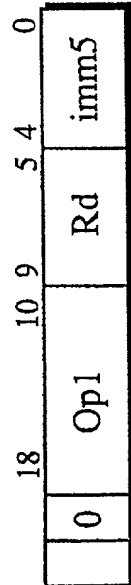


FIG. 2B

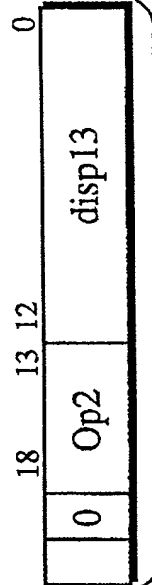


FIG. 2C

21 BITS

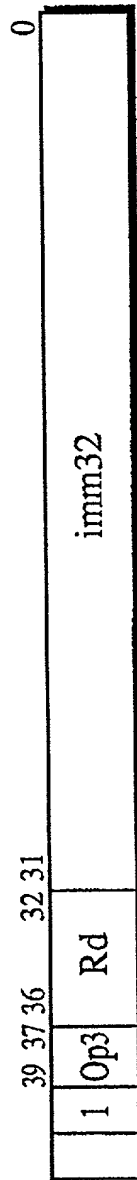


FIG. 2D

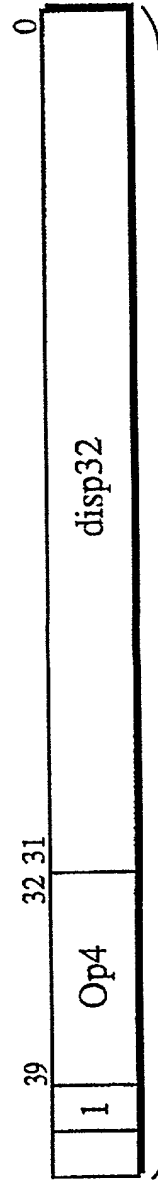


FIG. 2E

42 BITS

FIG. 3A

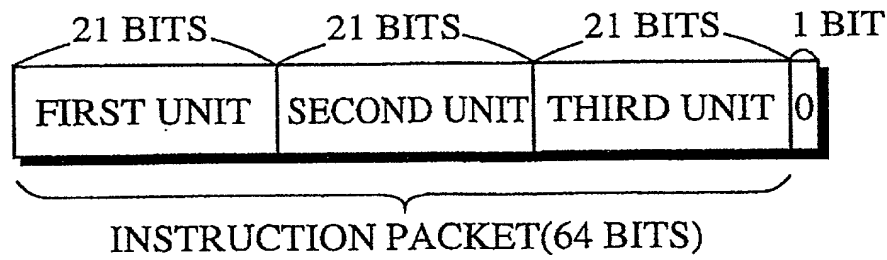


FIG. 3B

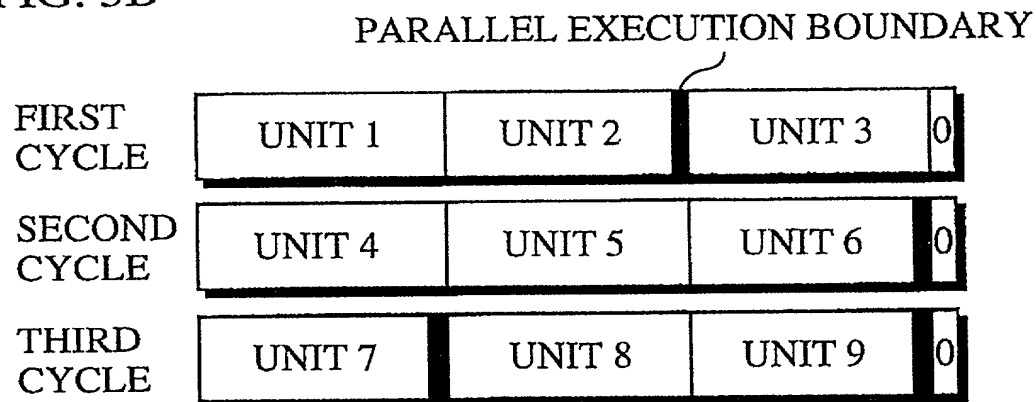


FIG. 3C

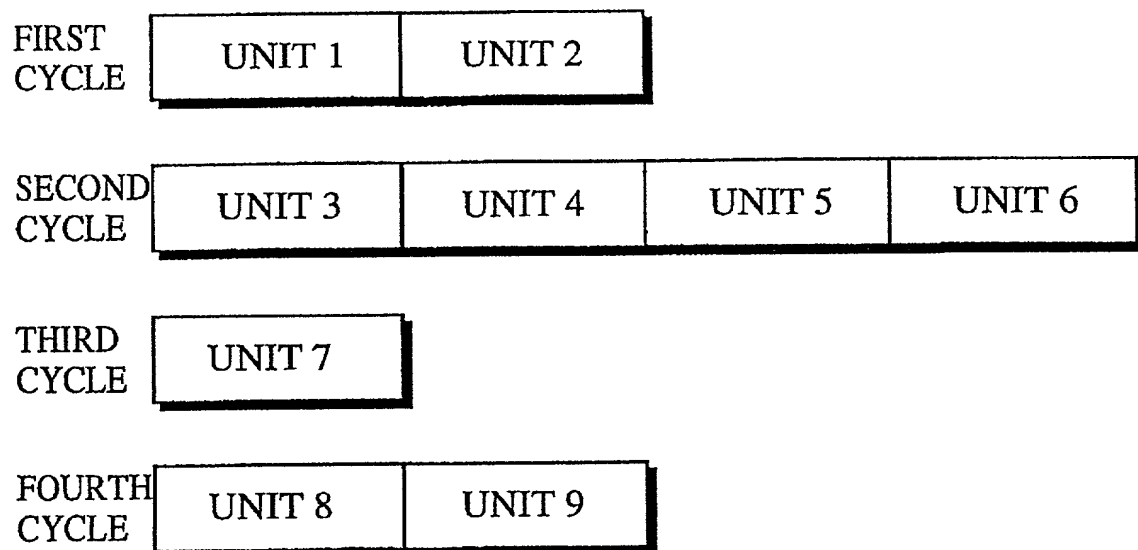


FIG. 4

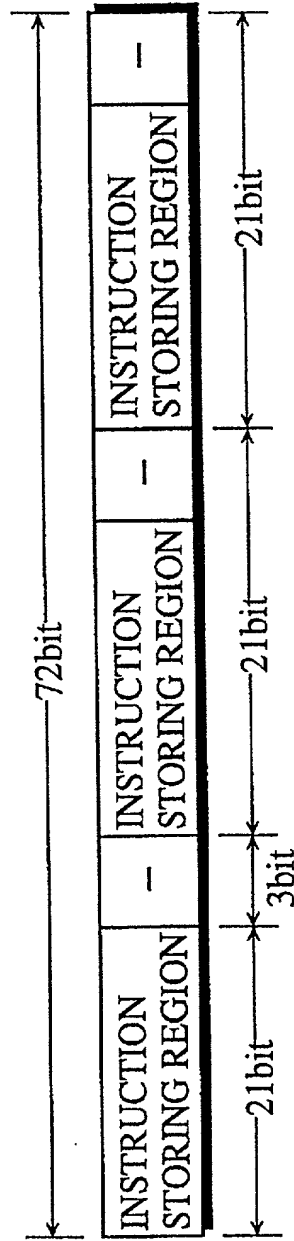


FIG. 5

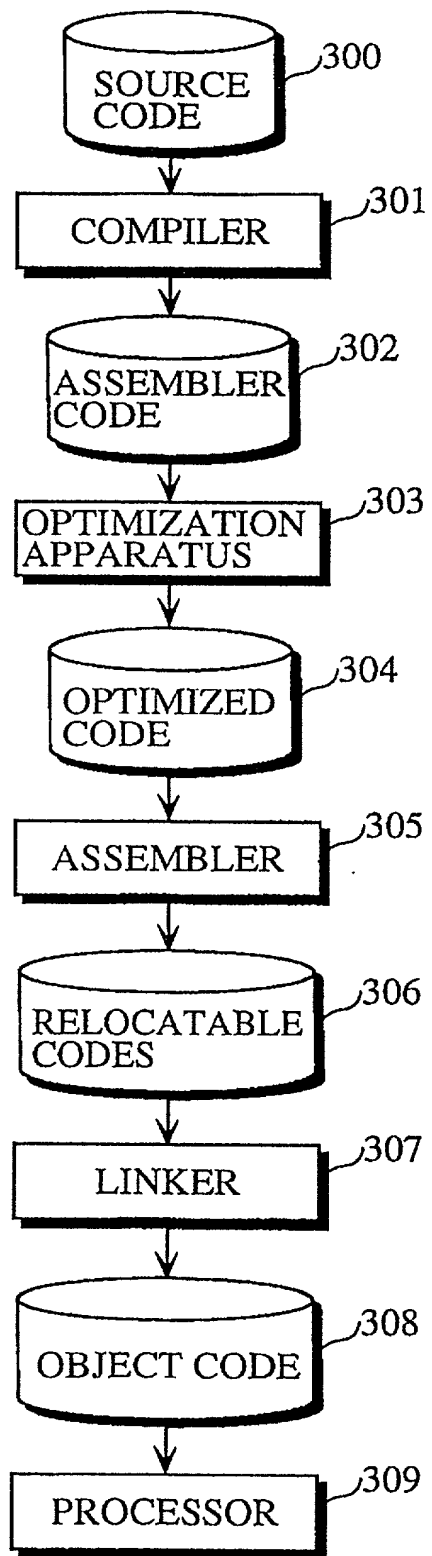


FIG. 6

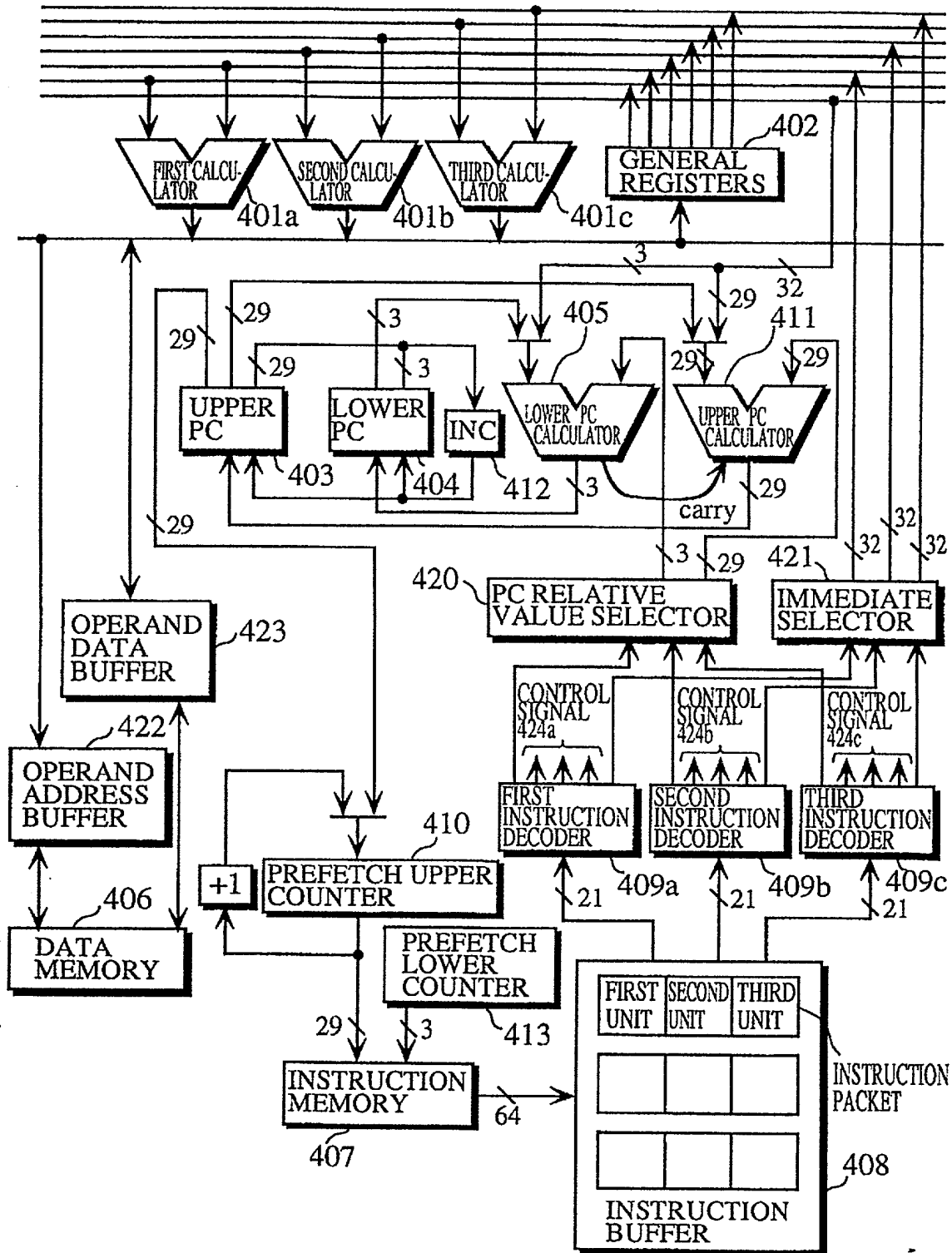


FIG. 7

IN-PACKET ADDRESS BEFORE UPDATING	3'b000	3'b010	3'b100
INCREMENT VALUE			
1	3'b010	3'b100	3'b000 (CARRY 1)
2	3'b100	3'b000 (CARRY 1)	3'b010 (CARRY 1)
3	3'b000 (CARRY 1)	3'b010 (CARRY 1)	3'b100 (CARRY 1)
4	3'b010 (CARRY 1)	3'b100 (CARRY 1)	3'b000 (CARRY 2)

FIG. 8A

LOWER 3 BITS OF ADDRESS VALUE LOWER 3 BITS OF PC RELATIVE VALUE	3'b000	3'b010	3'b100
3'b000	3'b000	3'b010	3'b100
3'b010	3'b010	3'b100	3'b000 (CARRY 1)
3'b100	3'b100	3'b000 (CARRY 1)	3'b010 (CARRY 1)

FIG. 8B

LOWER 3 BITS OF ADDRESS VALUE(BEFORE SUBTRACTION)	LOWER 3 BITS OF ADDRESS VALUE(TO BE SUBTRACTED)	3'b00	0b010	0b100
3'b000	3'b000	3'b000	3'b100 (CARRY 1)	3'b010 (CARRY 1)
3'b010	3'b010	3'b010	3'b000	3'b100 (CARRY 1)
3'b100	3'b100	3'b100	3'b010	3'b000

FIG. 9

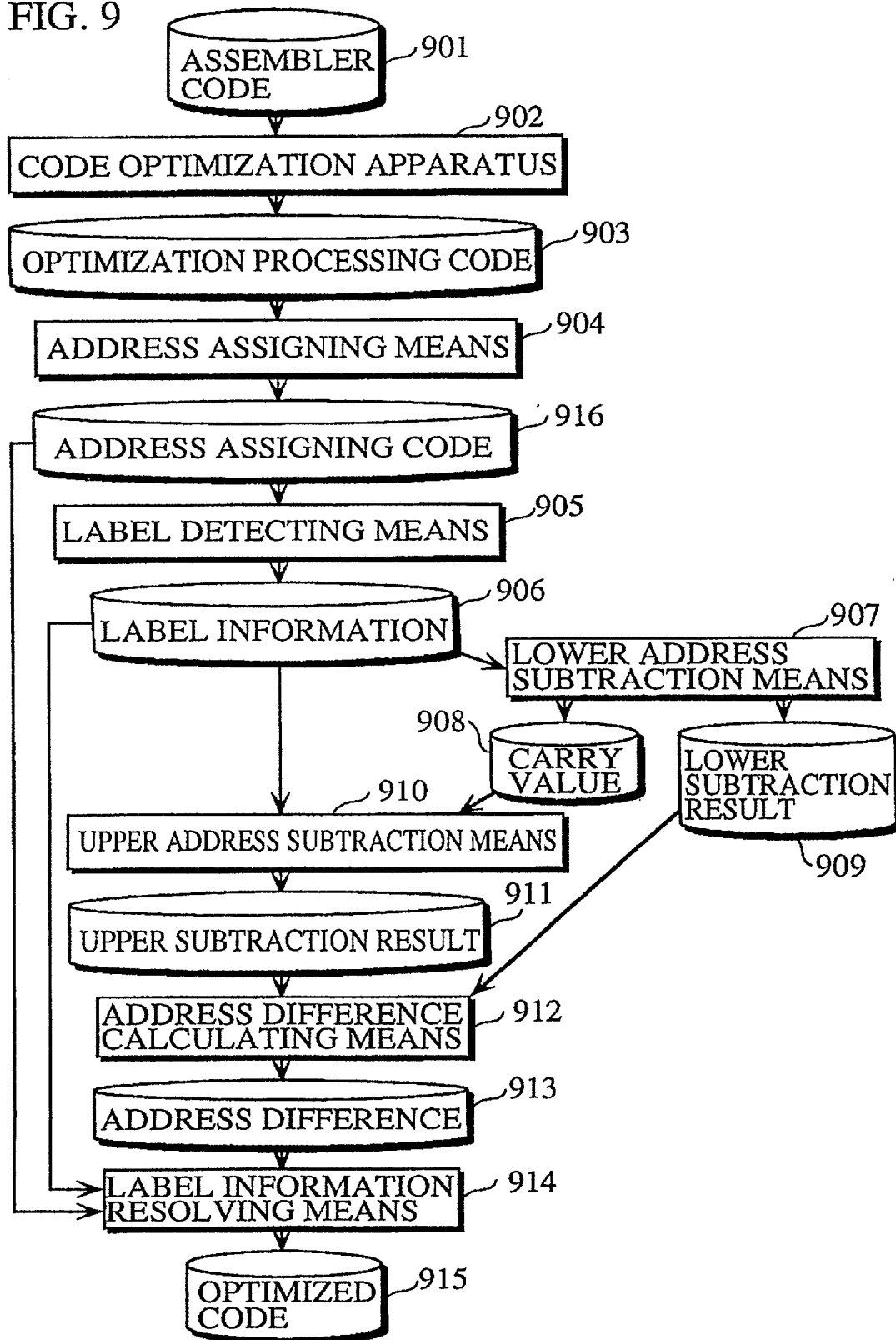


FIG. 10

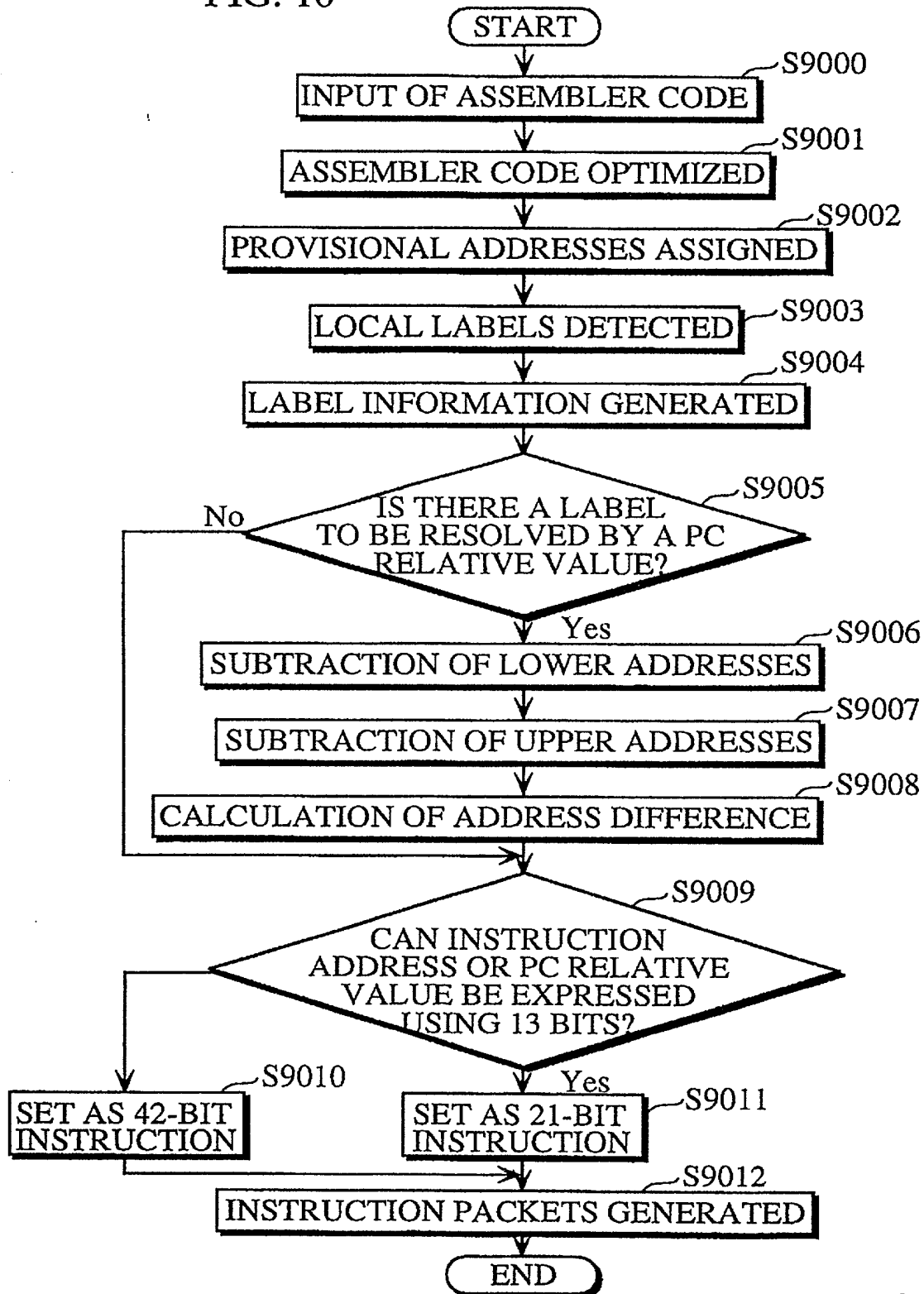


FIG. 11

L1: mov r2, r1	· · 1000
jsr f	· · 1001
add r0, r4	· · 1002
and r1, r3	· · 1003
mov L2, r2	· · 1004
ld (r2), r0	· · 1005
bra L1	· · 1006
add r2, r3	· · 1007
...	
L2: ...	· · 1008

FIG. 12

32'h00000800	L1: mov r2, r1	· · 1000
32'h00000802	jsr f	· · 1001
32'h00000804	add r0, r4	· · 1002
32'h00000808	and r1, r3	· · 1003
32'h0000080a	mov L2, r2	· · 1004
32'h00000810	ld (r2), r0	· · 1005
32'h00000812	bra L1	· · 1006
32'h00000814	add r2, r3	· · 1007
	...	
32'h12345678	L2: ...	· · 1008

FIG. 13

INSTRUCTION	RESOLVING VALUE
mov L2, r2	ADDRESS 32'h12345678
bra L1	PC RELATIVE VALUE 32'h00000800-32'h00000812

FIG. 14

L1:	mov r2, r1		jsr f		add r0, r4	· · 1300
	and r1, r3		mov L2, r2		(mov L2, r2)	· · 1301
	ld (r2) , r0		bra L1		add r2, r3	· · 1302
	...					
L2:						· · 1303

FIG. 15

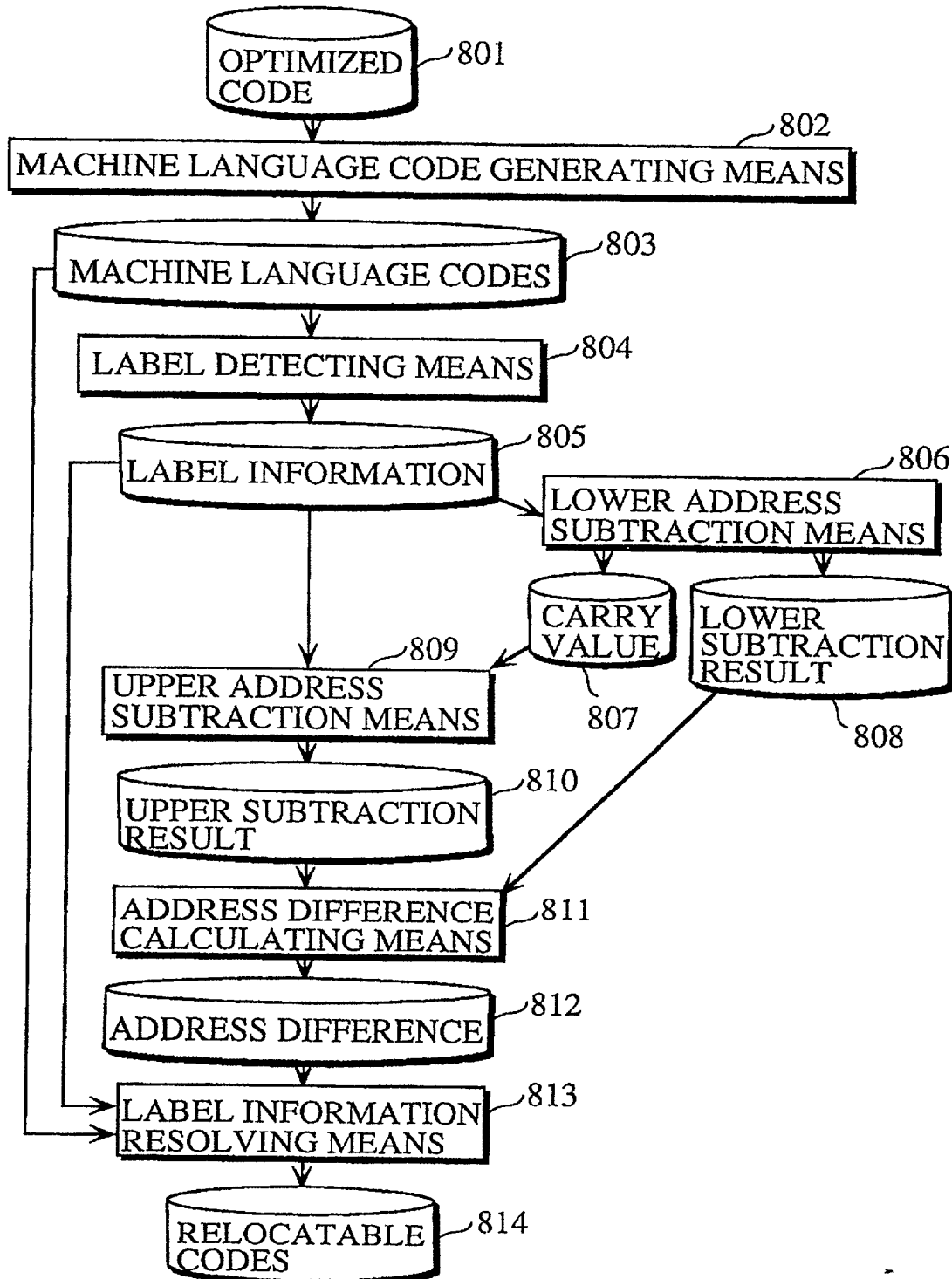


FIG. 16

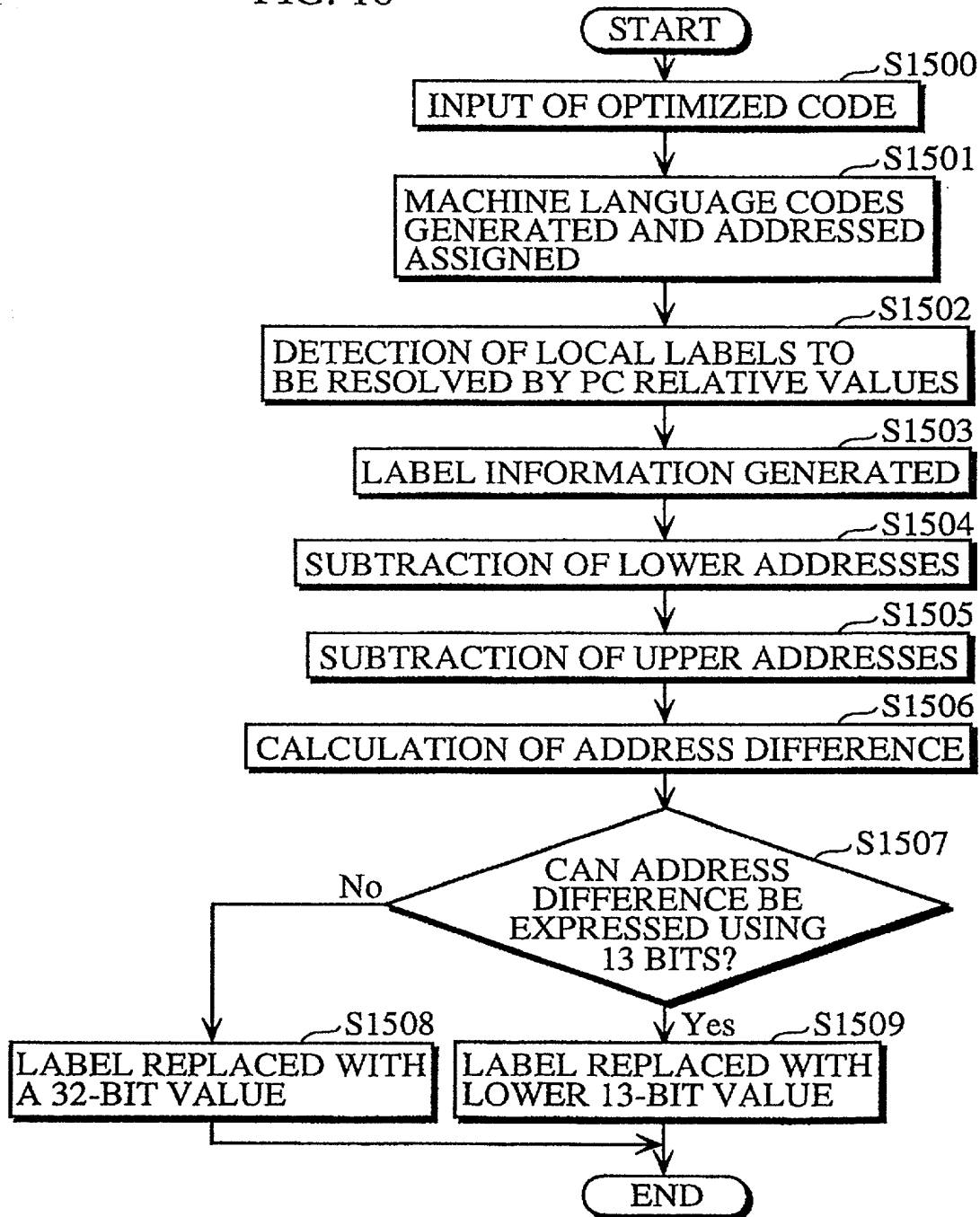


FIG. 17

29'h00000000	L1: mov r2, r1 ...1401	jsr f	...1402	add r0, r4 ...1403	. . 1400
29'h00000001	and r1, r3 ...1405	mov L2, r2	...1406		. . 1404
29'h00000002	ld (r2), r0 ...1408	bra L1	...1409	add r2, r3 ...1410	. . 1407
...					
29'h02468acf	L2: 1411

TOP OF PAGE 4001

FIG. 18

INSTRUCTION	RESOLVING VALUE
bra L1	PC RELATIVE VALUE 32'h00000000-32'h00000012

FIG. 19

PARALLEL EXECUTION BOUNDARY INFORMATION		BIT FORMAT INFORMATION		UNUSED BIT AREA	
29'h00000000	0:0:L1: mov r2, r1 ... 1601	1:0:jsr f	...1602	0:0: add r0, r4 ... 1603	0 ... 1600
29'h00000001	0:0: and r1, r3 ... 1605	1:1:mov L2, r2	...1606	0	0 ... 1604
29'h00000002	0:0: ld (r2), r0 ... 1608	1:0:bra 13'h1fec...1609	0:0: add r2, r3 ... 1610		0 ... 1607
...					
29'h02468acf	L2:	...		0	0 ... 1611

FIG. 20

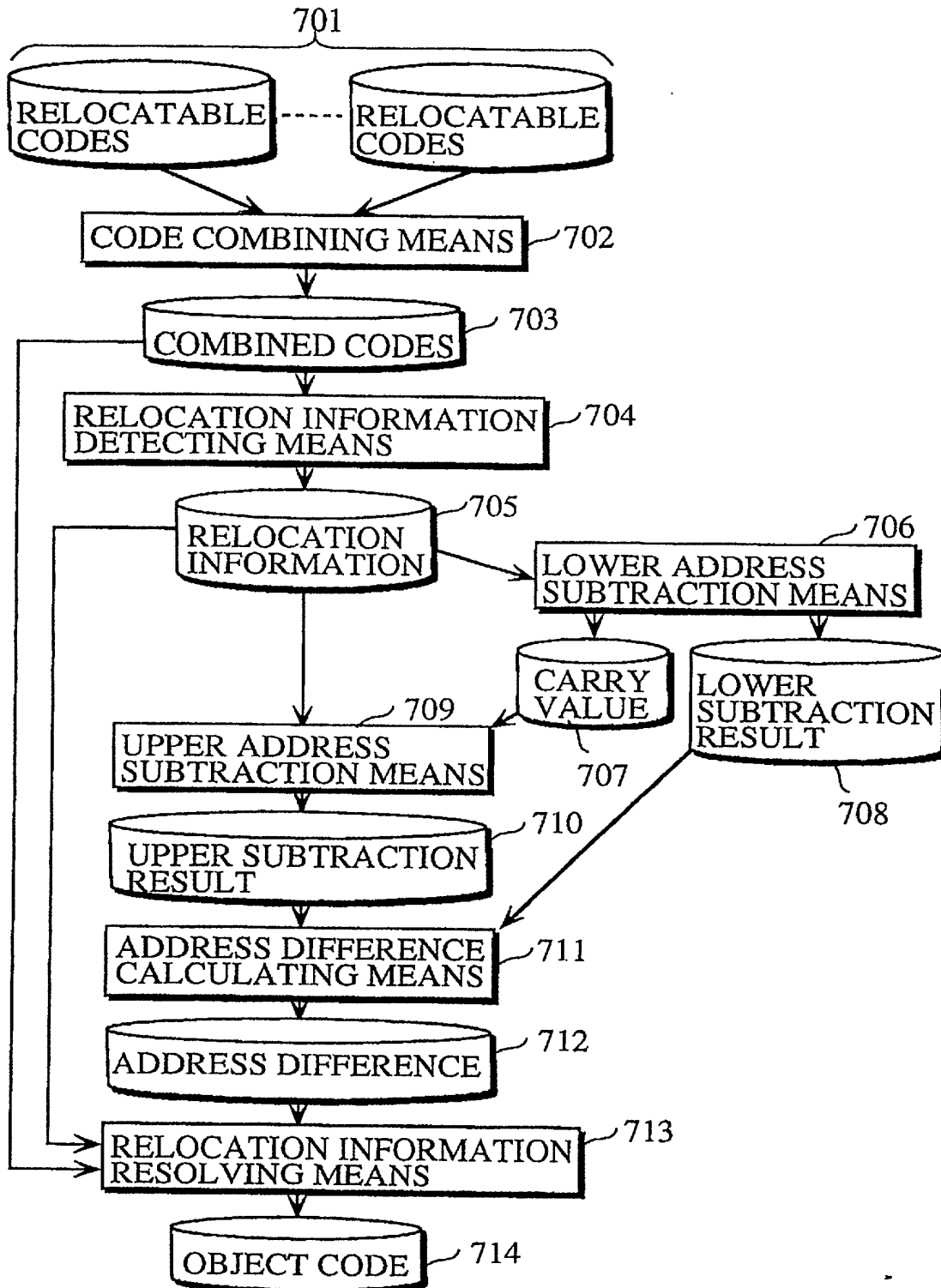


FIG. 20 "SEE FIG. 20"

FIG. 21

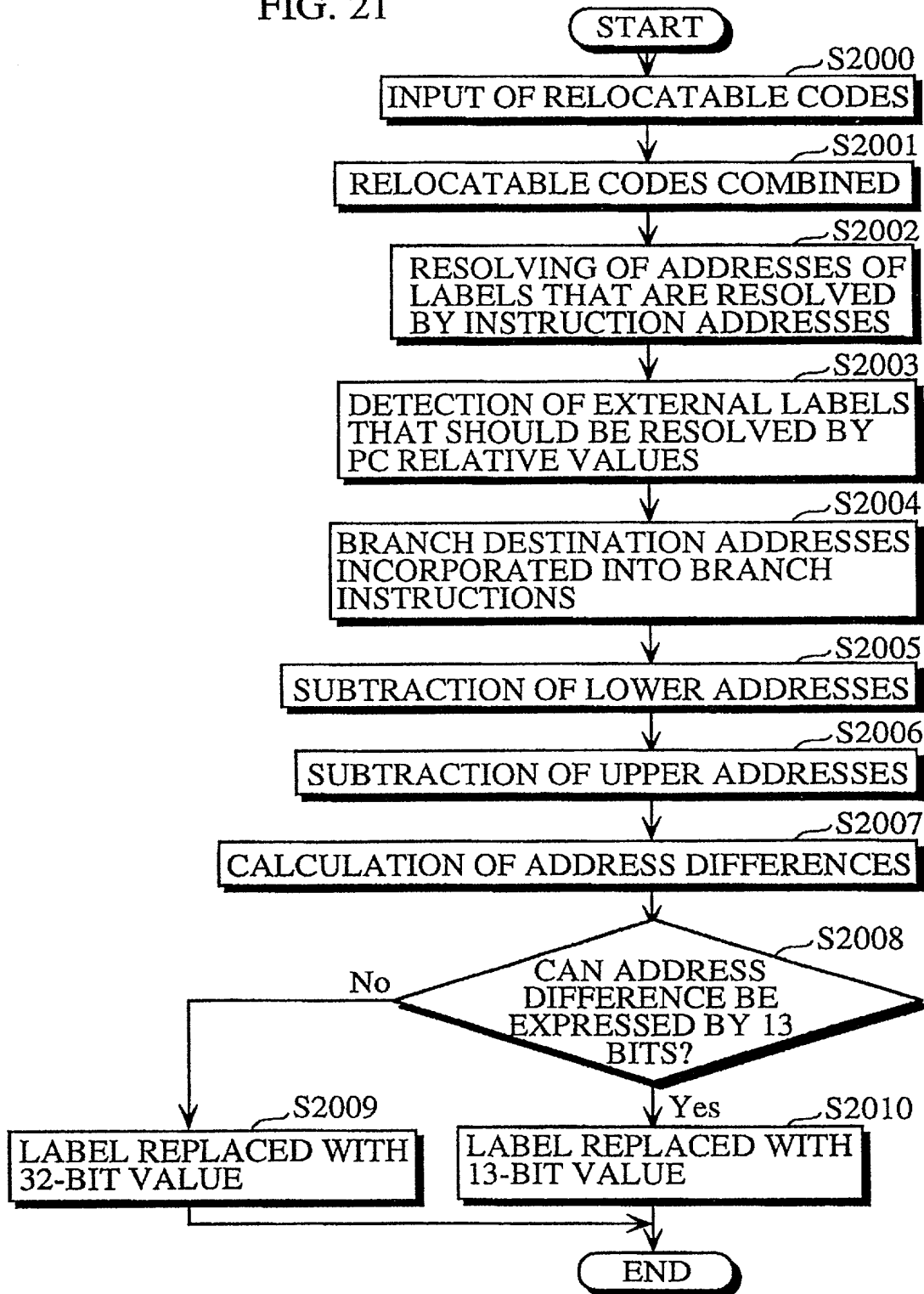


FIG. 22



FIG. 23

29'h00000000	0:0:f:	ret	...1801	0:0:nop	...1802	1:0:nop	...1803	0	· · 1800
29'h00000001	0:0:L1:	mov r2, r1	...1805	1:0:jsr f	...1806	0:0: add r0, r4	...1807	0	· · 1804
29'h00000002	0:0:	and r1, r3	...1809	1:1:mov L2, r2	...1810			0	· · 1808
29'h00000003	0:0:	ld (r2), r0	...1812	1:0:bra 13'h1fec	...1813	0:0: add r2, r3	...1814	0	· · 1811
...									
29'h02468ad0	L2:		...					0	· · 1815

FIG. 24

29'h00000000	0:0:f:	ret	...1901	0:0:nop	...1902	1:0:nop	...1903	0	· · 1900	
29'h00000001	0:0:L1:	mov r2, r1	...1905	1:0:jsr f	...1906	0:0: add r0, r4	...1907	0	· · 1904	
29'h00000002	0:0:	and r1, r3	...1909	1:1:mov 32'h12345680, r2	...1910			0	· · 1908	
29'h00000003	0:0:	ld (r2), r0	...1912	1:0:bra 13'h1fec...	1913	0:0: add r2, r3	...1914	0	· · 1911	
...										
29'h02468ad0	L2:	...							0	· · 1915

FIG. 25

INSTRUCTION	RESOLVING VALUE
jsr f	PC RELATIVE VALUE 32'h00000000-32'h0000000a

FIG. 26

29'h00000000	0:0:f	ret	...2101	0:0:nop	...2102	1:0:nop	...2103	0	· · 2100
29'h00000001	0:0:L1	mov r2, r1	...2105	1:0:jsr 13'h1ff4	...2106	0:0: add r0, r4	...2107	0	· · 2104
29'h00000002	0:0	and r1, r3	...2109	1:1:mov 32'h12345680, r2	...2110			0	· · 2108
29'h00000003	0:0	ld (r2), r0	...2112	1:0:bra 13'h1fec	...2113	0:0: add r2, r3	...2114	0	· · 2111
...									
29'h02468ad0	L2:		...					0	· · 2115

FIG. 28A

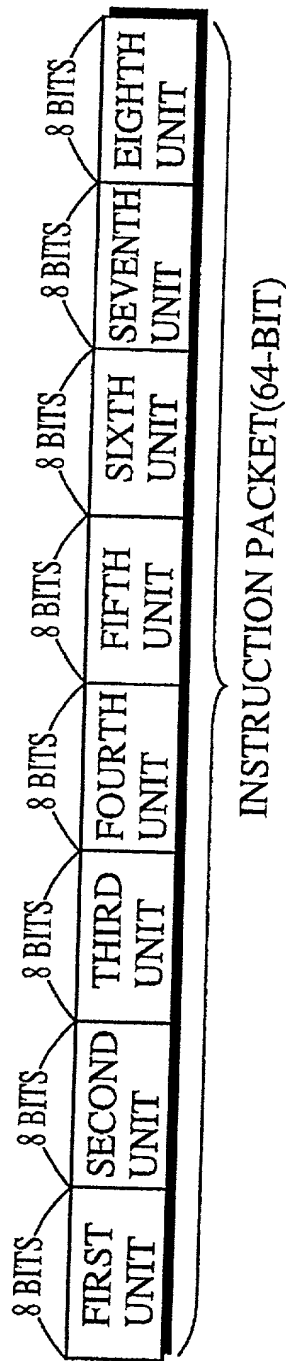


FIG. 28B

2-UNIT INSTRUCTION

3-UNIT INSTRUCTION

5-UNIT INSTRUCTION

6-UNIT INSTRUCTION

FIG. 28C

IN-PACKET ADDRESS	UNIT
3'b000	FIRST UNIT
3'b001	SECOND UNIT
3'b010	THIRD UNIT
3'b011	FOURTH UNIT
3'b100	FIFTH UNIT
3'b101	SIXTH UNIT
3'b110	SEVENTH UNIT
3'b111	EIGHTH UNIT

FIG. 29A

LOWER 3 BITS OF PC RELATIVE VALUE	LOWER 3 BITS OF ADDRESS VALUE		
	3'b000	3'b010	3'b100
3'b000	3'b000	3'b010	3'b100
3'b010	3'b010	3'b100	3'b000 (CARRY IGNORED)
3'b100	3'b100	3'b000 (CARRY IGNORED)	3'b010 (CARRY IGNORED)

FIG. 29B

LOWER 3 BITS OF ADDRESS VALUE (BEFORE BE SUBTRACTION)	LOWER 3 BITS OF ADDRESS VALUE (TO BE SUBTRACTED)		
	3'b000	3'b010	3'b100
3'b000	3'b000	3'b100 (CARRY IGNORED)	3'b010 (CARRY IGNORED)
3'b010	3'b010	3'b000	3'b100 (CARRY IGNORED)
3'b100	3'b100	3'b010	3'b000

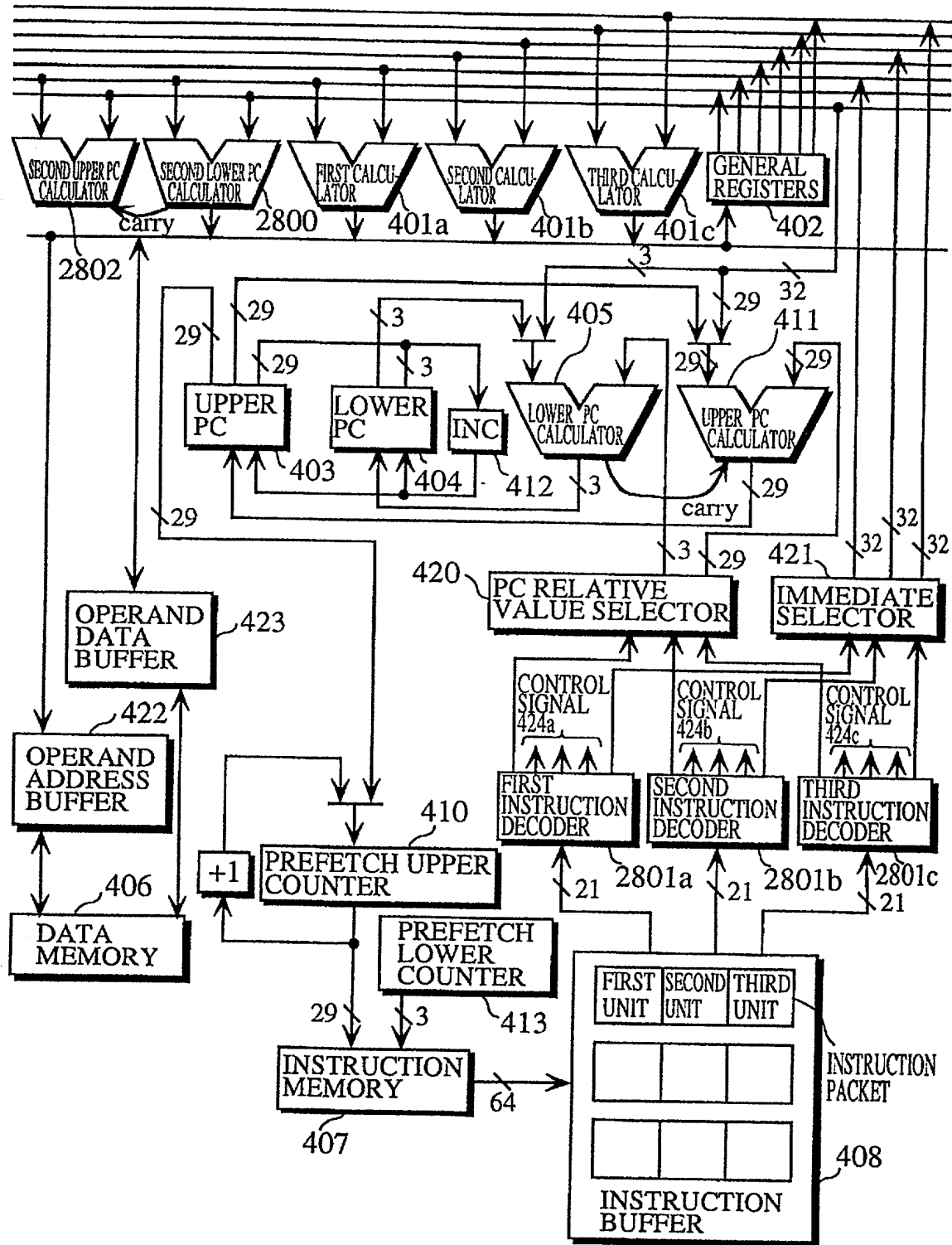
FIG. 31A

LOWER 3 BITS OF PC RELATIVE VALUE	LOWER 3 BITS OF ADDRESS VALUE		
	3'b000	3'b010	3'b100
3'b000	3'b000	3'b000	3'b000
3'b010	3'b010	3'b010	3'b010
3'b100	3'b100	3'b100	3'b100

FIG. 31B

LOWER 3 BITS OF PC RELATIVE VALUE	LOWER 3 BITS OF ADDRESS VALUE		
	3'b000	0b010	0b100
3'b000	3'b000	3'b000	3'b000
3'b010	3'b010	3'b010	3'b010
3'b100	3'b100	3'b100	3'b100

FIG. 34



	MNEMONIC	OPERATION
FIG. 35A	addpc disp , Rn	$Rn + disp \rightarrow Rn$
FIG. 35B	subpc disp , Rn	$Rn - disp \rightarrow Rn$

FIG. 36

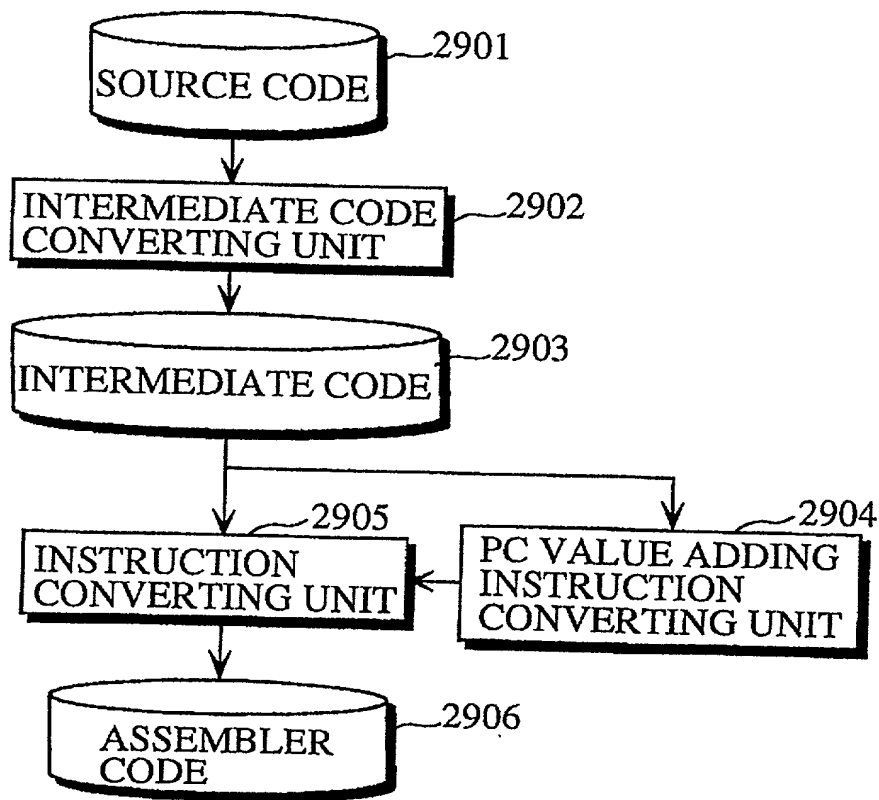


FIG. 37

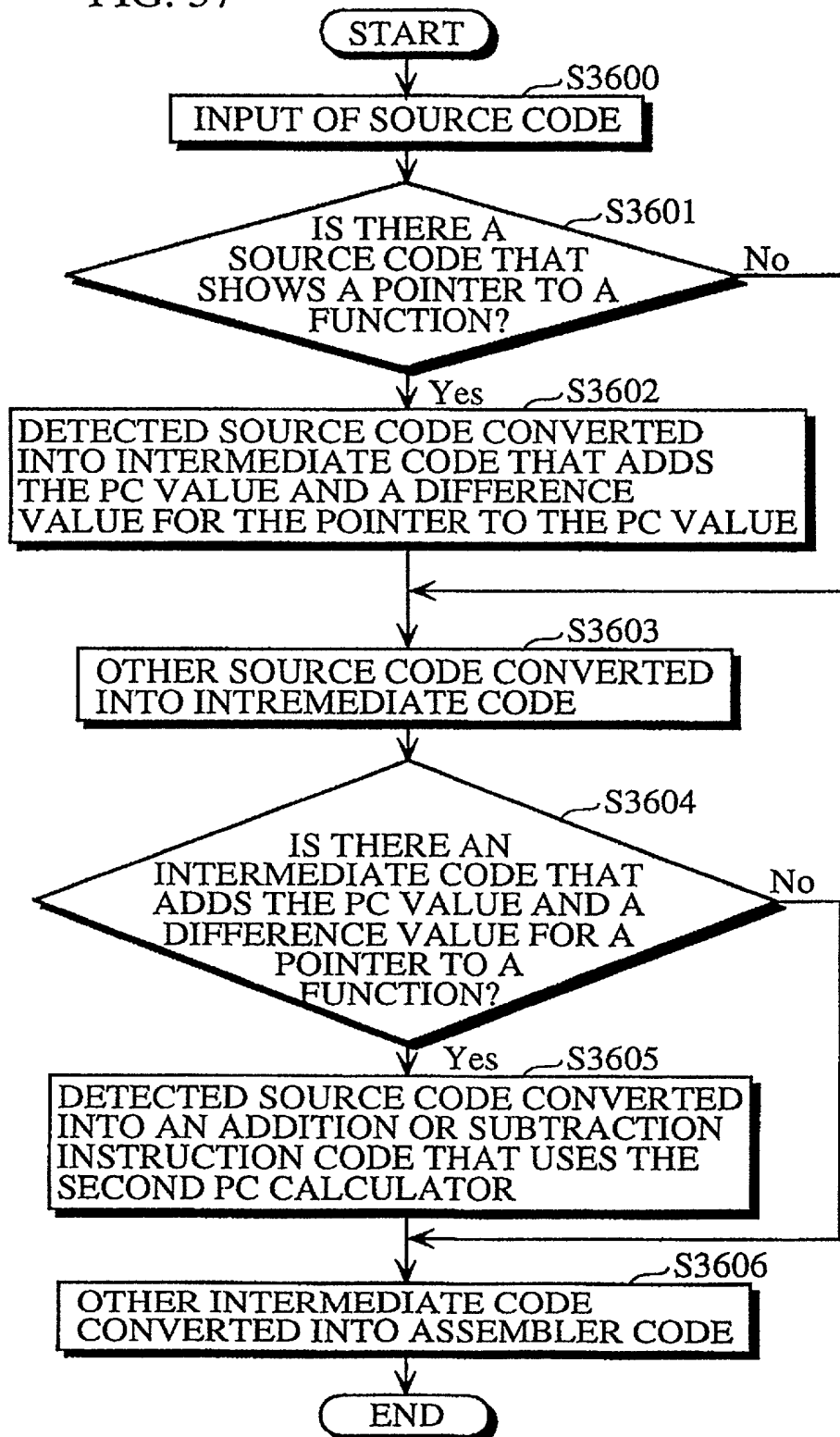


FIG. 38

```
extern int g1 ();
extern int g2 ();
extern int g3 ();
extern int g4 ();

f (int i)
{
    int (*fp) ();

    switch(i) {
        case 1 : fp = g1 ;
                break ;
        case 2 : fp = g2 ;
                break ;
        case 3 : fp = g3 ;
                break ;
        default : fp = g4 ;
    }

    (*fp) () ;
}
```

FIG. 39

f:	tmp = PC	3201
	i != 1	3202
	br L1	3203
	fp = (g1 - f) + tmp	3204
	jmp L	3205
L1:	i != 2	3206
	br L2	3207
	fp = (g2 - f) + tmp	3208
	jmp L	3209
L2:	i != 3	3210
	br L3	3211
	fp = (g3 - f) + tmp	3212
	jmp L	3213
L3:	fp = (g4 - f) + tmp	3214
L:	*(fp) (i)	3215

FIG. 40

f :	mov	PC, r1	3201
	compne	1, r0	3202
	br	L1	3203
	addpc	g1-f, r1	3204
	jmp	L	3205
L1 :	cmpne	2, r0	3206
	br	L2	3207
	addpc	g2-f, r1	3208
	jmp	L	3209
L2 :	cmpne	3, r0	3210
	br	L3	3211
	addpc	g3-f, r1	3212
	jmp	L	3213
L3 :	addpc	g4-f, r1	3214
L :	jsr	(r1)	3215
	ret		3216

FIG. 42

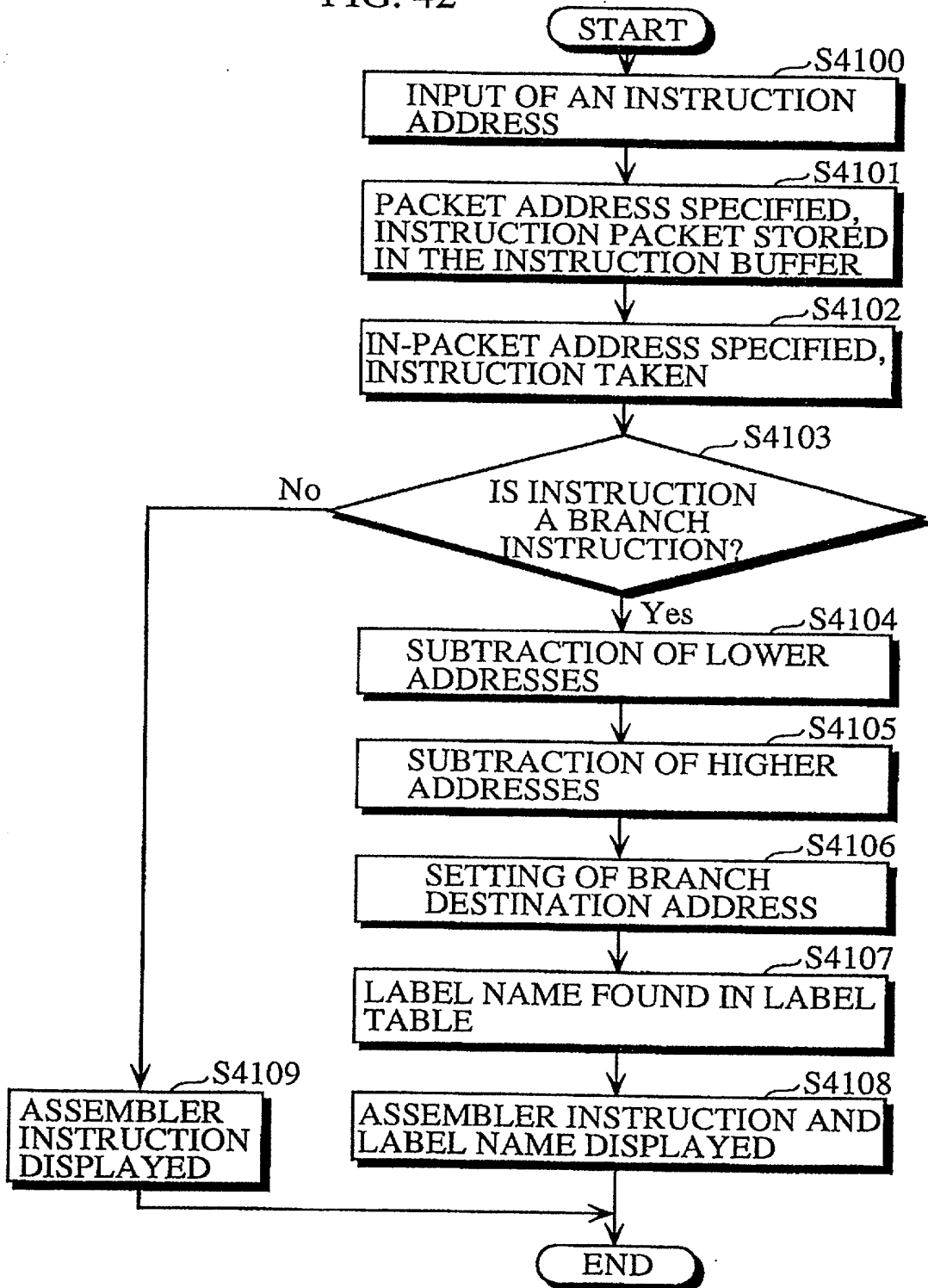


FIG. 43

